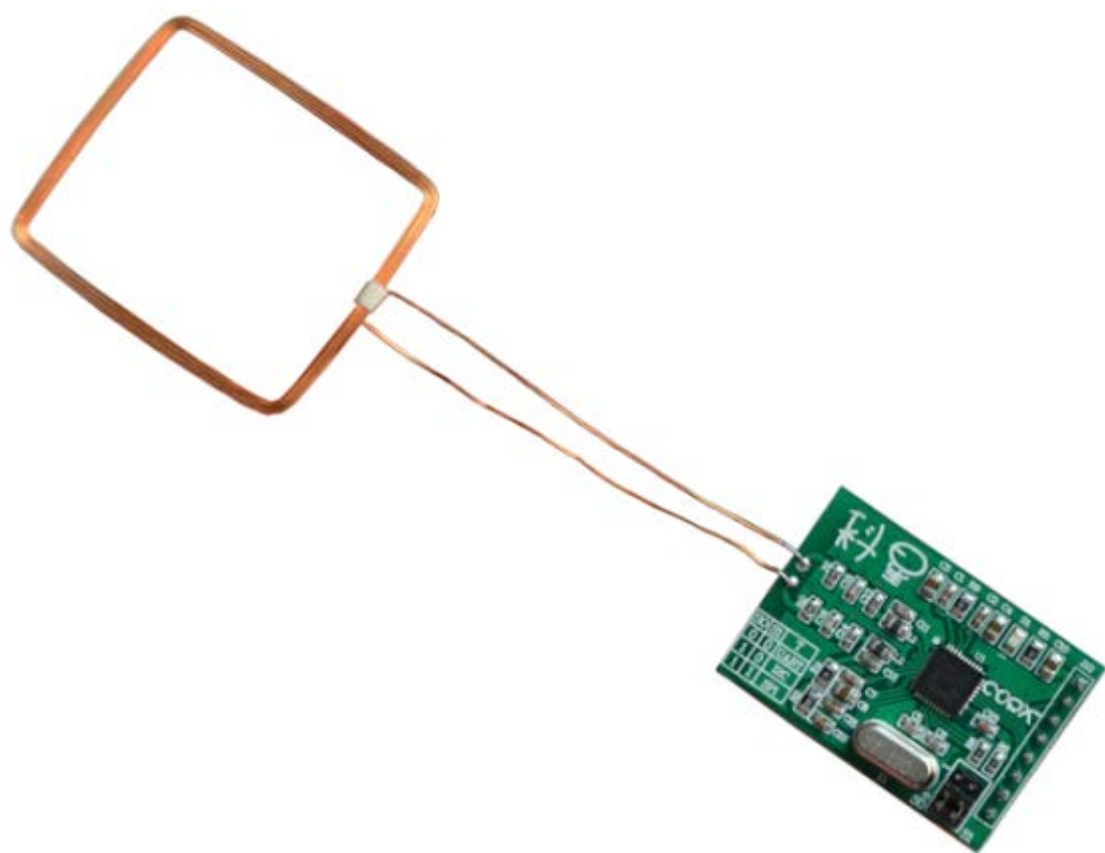


科星 NFC 模块用户手册



NFC 模块概述

NFC-M02 是我公司最新研发生产的一款基于 NXP 的 PN532 芯片体积小、低成本、高性能的 NFC 读写 Mifare 卡的模块，它可以轻松的用短路帽实现 UART、SPI 和 I2C 接口间的切换，以适用更多的硬件平台。大大的缩减了客户的研发成本和时间，使产品快速投入使用，一鸣惊人。

NFC 模块各项参数如下表：

表一.NFC 模块参数列表

适用范围	支持 Mifare 系列兼容卡
对外接口	UART / SPI / I2C（用户可通过跳帽选择）
核心芯片	PN532
工作频率	13.56 MHz
供电电压	3.3V
内核	80C51
ROM	40 K
RAM	1 K
有效距离	35mm
模块尺寸	25*37mm
天线尺寸	34*39mm
工作温度	-15~75℃
储藏温度	-25~85℃

一、Mifare 卡介绍

1.1、Mifare 卡概述

MIFARE 卡是目前世界上使用量最大、技术最成熟、性能最稳定、内存容量最大的一种感应式智能 IC 卡。至于“Mifare”这个名字的由来，据说 1998 年 Philips 收购了瑞士的米克隆(Mikron)公司，该公司之前开发了一套收费系统叫作 Mikron FARE-collection System，即米克隆收费系统，简称为 Mifare。

MIFARE 是 Philips Electronics 所拥有的 13.56MHz 非接触性辨识技术。Philips 并没有制造卡片或卡片阅读器，而是在开放的市场上贩售相关技术与芯片，卡片和卡片阅读器之制造商再利用它们的技术来创造独特的产品给一般使用者。

MIFARE 经常被认为是一种智能卡的技术，这是因为它可以在卡片上兼具读写的功能。事实上，MIFARE 仅具备记忆功能，必须搭配处理器卡才能达到读写功能。

MIFARE 的非接触式读写功能是设计来处理大众运输系统中的付费交易部分，其与众不同的地方是具备执行升幂和降序的排序功能，简化资料读取的过程。尽管接触性智能卡也能够执行同样的动作，但非接触性智能卡的速度更快且操作更简单，而且卡片阅读器几乎不需要任何维修，卡片也较为耐用。

1.2、Mifare 卡特征

1.2.1 MIFARE 接口 ISO/IEC 14443A

- ☐ 无线传送数据和能量不需要电池
- ☐ 工作距离最高可达 100mm 由天线的结构 geometry 决定
- ☐ 工作频率 13.56MHz,

卡片与读写器之间通讯的数据速率有 4 种:106Kbps,212Kbps,424Kbps,847Kbps。但在读卡选择命令(含)之前,通讯速率只能是 106Kbps。读卡选择之后,卡片和读写器可以协商使用什么样的速率。106Kbps 是怎么来的,它是载波频率 13.56MHz 除以 128 得来的,通俗的说法是“载波 128 分频”。

- ☐ 数据传送速度快 106kbit/s
- ☐ 数据高度可靠正确 16 位 CRC 奇偶校验位编码位计数
- ☐ 真正的反冲突
- ☐ 典型的购票处理 ticketing transaction <100ms 包括备份管理

1.2.2 EEPROM

- ☐ 1K 字节分成 16 个区，每区又分成 4 段，每一段中有 16 个字节
- ☐ 用户可以定义每一个存储器段的访问条件
- ☐ 数据可以保持 10 年
- ☐ 可写 100,000 次

1.3、Mifare 卡类型

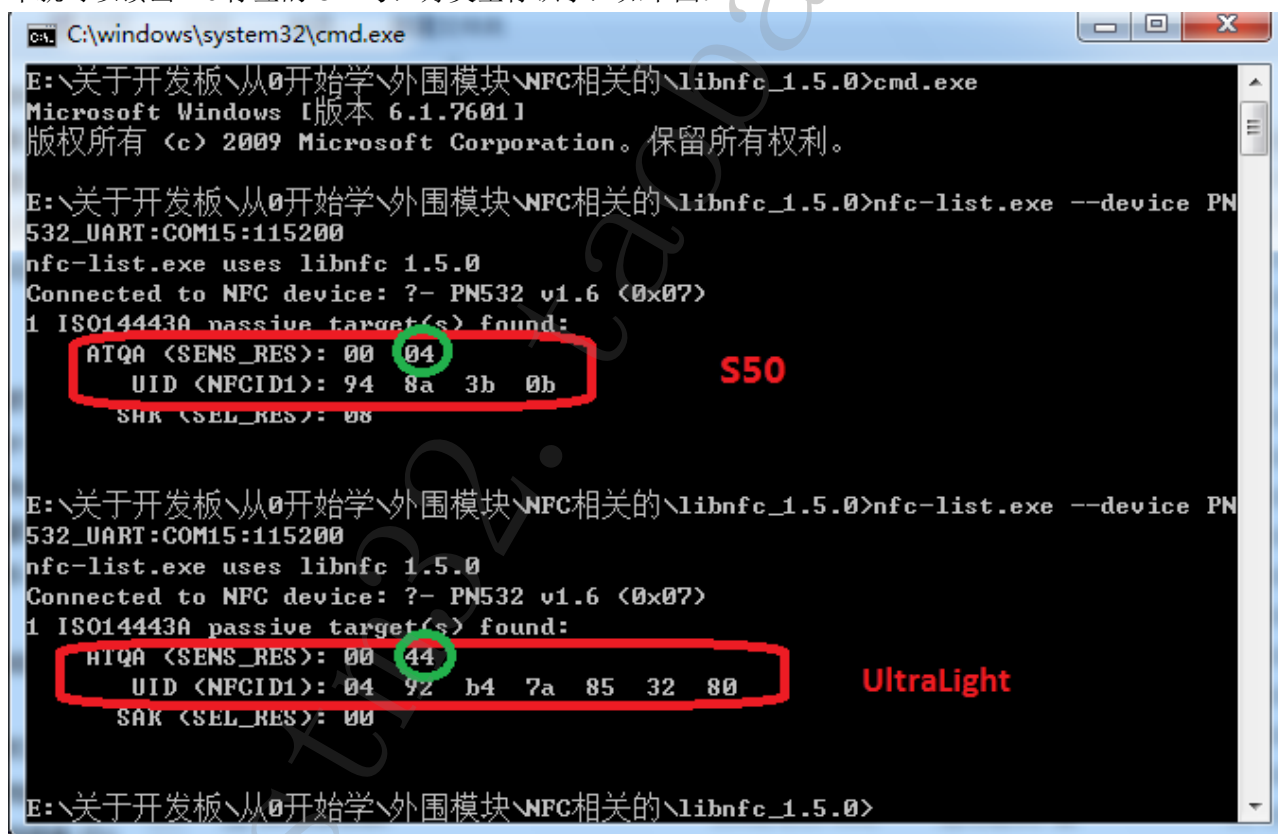
Mifare 是 NXP 公司生产的一系列遵守 ISO14443A 标准的射频卡, 包 Mifare S50、Mifare S70、Mifare UltraLight、Mifare Pro、Mifare Desfire 等, 由于 Mifare 的巨大影响力, 业内有时把其它公司生产的遵守 ISO14443A 标准的射频卡也称为“Mifare”, 尤其是 Mifare S50 卡片, 几乎就是 ISO14443A 标准的代言人。

Mifare 系列卡片有时也根据卡内使用芯片的不同, 把 Mifare UltraLight 称为 MF0, Mifare S50 和 S70 称为 MF1, Mifare Pro 称为 MF2, Mifare Desfire 称为 MF3。

这些卡片都有一个全球唯一的序列号, 序列号的长度可能是 4 字节, 7 字节或 10 字节。S50 是 4 字节, UltraLight 是 7 字节。

识别不同卡的类型可以使用“libnfc_1.5.0”里面的程序, 演示如下:

打开运行, 输入 cmd, 进入到该文件夹的目录下, 输入: “nfc-list.exe --device PN532_UART:COM15:115200”, 端口号根据自己电脑的设备管理器的信息改一下。然后敲回车就可以读出 nfc 标签的 UID 号, 好类型标识了, 如下图:



```
C:\windows\system32\cmd.exe
E:\关于开发板\从0开始学\外围模块\NFC相关的\libnfc_1.5.0>cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

E:\关于开发板\从0开始学\外围模块\NFC相关的\libnfc_1.5.0>nfc-list.exe --device PN532_UART:COM15:115200
nfc-list.exe uses libnfc 1.5.0
Connected to NFC device: ?- PN532 v1.6 (0x07)
1 ISO14443A passive target(s) found:
  ATQA (SENS_RES): 00 04
  UID (NFCID1): 94 8a 3b 0b
  SAK (SEL_RES): 00
  S50

E:\关于开发板\从0开始学\外围模块\NFC相关的\libnfc_1.5.0>nfc-list.exe --device PN532_UART:COM15:115200
nfc-list.exe uses libnfc 1.5.0
Connected to NFC device: ?- PN532 v1.6 (0x07)
1 ISO14443A passive target(s) found:
  ATQA (SENS_RES): 00 44
  UID (NFCID1): 04 92 b4 7a 85 32 80
  SAK (SEL_RES): 00
  UltraLight

E:\关于开发板\从0开始学\外围模块\NFC相关的\libnfc_1.5.0>
```

ATQA 有两个字节, 第一个字节的值没有规定(RFU), 第二个字节的高两位 b7b6 表示卡序列号长度(“00”为 4 字节, “01”为 7 字节, “10”为 10 字节), b5 位的值没有规定(RFU), b4-b0 表示是否遵守面向比特的防冲突机制, 如果遵守, b4-b0 必须有且仅有 1 位为 1。通常情况下, Mifare S50 的 ATQA 是 0004H, Mifare S70 的 ATQA 是 0002H, Mifare UltraLight 的 ATQA 是 0044H, Mifare Light 的 ATQA 是 0010H, Mifare Desfire 的 ATQA 是 0344H。业内习惯称 ATQA 为卡类型, 并且称“Mifare S50 的卡类型是 0004H, Mifare S70 的卡类型是 0002H, Mifare UltraLight 的卡类型是 0044H, Mifare Desfire 的卡类型是 0344H.....”, 这种说法其实是不严谨的。已经出现了 ATQA 为 0044H 和 0344H 的卡片, 但这种卡片并不是 Mifare UltraLight 和 Mifare Desfire, 而是一种新的 7 字节的 Mifare S50。这很正常, 因为 ISO14443A 中规定, ATQA 的作

用是卡片表明自己是否遵守面向比特的防冲突机制以及自身卡序列号的长度,并不是表示哪种类型的卡片。

上图中是读出的是 S50 和 Mifare UltraLight 这两种卡的类型信息和卡的 UID 号。

1.3.1 Mifare S50 和 S70

Mifare S50 和 Mifare S70 又常被称为 Mifare Standard、Mifare Classic、MF1, 是遵守 ISO14443A 标准的卡片中应用最为广泛、影响力最大的一员。而 Mifare S70 的容量是 S50 的 4 倍, S50 的容量是 1K 字节, S70 的容量为 4K 字节。读写器对卡片的操作时序和操作命令, 二者完全一致。

Mifare S50 和 Mifare S70 的每张卡片都有一个 4 字节的全球唯一序列号, 卡上数据保存期为 10 年, 可改写 10 万次, 读无限次。一般的应用中, 不用考虑卡片是否会被读坏写坏的问题, 当然暴力硬损坏除外。

Mifare S50 和 Mifare S70 的区别主要有两个方面。一是读写器对卡片发出请求命令, 二者应答返回的卡类型(ATQA)字节不同。Mifare S50 的卡类型(ATQA)是 0004H, Mifare S70 的卡类型(ATQA)是 0002H。另一个区别就是二者的容量和内存结构不同。

Mifare S50 把 1K 字节的容量分为 16 个扇区(Sector0-Sector15), 每个扇区包括 4 个数据块(Block0-Block3, 我们也将 16 个扇区的 64 个块按绝对地址编号为 0~63), 每个数据块包含 16 个字节(Byte0-Byte15), $64 \times 16 = 1024$ 。

Mifare S70 把 4K 字节的容量分为 40 个扇区(Sector0-Sector39), 其中前 32 个扇区(Sector0-Sector31)的结构和 Mifare S50 完全一样, 每个扇区包括 4 个数据块(Block0-Block3), 后 8 个扇区每个扇区包括 16 个数据块(Block0-Block15)。我们也将 40 个扇区的 256 个块按绝对地址编号为 0~255), 每个数据块包含 16 个字节(Byte0-Byte15), $256 \times 16 = 4096$ 。

1.3.2 Mifare UltraLight

Mifare UltraLight 又称为 MF0, 从 UltraLight(超轻的)这个名字就可以看出来, 它是一个低成本、小容量的卡片。低成本, 是指它是目前市场中价格最低的遵守 ISO14443A 协议的芯片之一; 小容量, 是指其存储容量只有 512bit(Mifare S50 有 8192bit)。

Mifare UltraLight 的 512bit 存储容量分成 16 个 Page, 每个 Page 包含 4 个字节, 如下图所示:

Byte Number	0	1	2	3	Page
Serial Number	SN0	SN1	SN2	BCC0	0
Serial Number	SN3	SN4	SN5	SN6	1
Internal / Lock	BCC1	Internal	Lock0	Lock1	2
OTP	OTP0	OTP1	OTP2	OTP3	3
Data read/write	Data0	Data1	Data2	Data3	4
Data read/write	Data4	Data5	Data6	Data7	5
Data read/write	Data8	Data9	Data10	Data11	6
Data read/write	Data12	Data13	Data14	Data15	7
Data read/write	Data16	Data17	Data18	Data19	8
Data read/write	Data20	Data21	Data22	Data23	9
Data read/write	Data24	Data25	Data26	Data27	10
Data read/write	Data28	Data29	Data30	Data31	11
Data read/write	Data32	Data33	Data34	Data35	12
Data read/write	Data36	Data37	Data38	Data39	13
Data read/write	Data40	Data41	Data42	Data43	14
Data read/write	Data44	Data45	Data46	Data47	15

Page0 和 Page1 以及 Page2 的第 1 个字节是卡片的 7 字节序列号及其校验字节，其中 $BCC0 = 0x88 \oplus SN0 \oplus SN1 \oplus SN2$ ， $BCC1 = SN3 \oplus SN4 \oplus SN5 \oplus SN6$ ，SN0 是制造商代码，由于 Mifare UltraLight 是 NXP 公司出品，因而 SN0 固定为 04H。Page2 的第 2 个字节 Internal 作为内部数据保留。以上共 10 个字节出厂时固化在存储区内，用户无法更改。

Page3 是一次性烧录(One Time Programmable,OTP)页,该页的内容在卡片出厂时全部被写为“0”，用户使用只能把某一位的内容写为“1”，而永远也不能把“1”写为 0，也就是说，新写入的 4 字节内容与卡内原来的内容进行异或，异或后的结果存储在卡片中。

Page4-Page15 是可读写的用户数据区，出厂时其内容初始化为 0，用户可以任意读写。

Page2 的第 3 和第 4 个字节用于将存储区锁定为只读。如下图所示，L4-L15 的某一位设置为 1，则对应序号的 Page 内容锁定为只读，每一个 Page 都可以单独设置。Lotp 用于锁定 Page3 为只读。“螳螂捕蝉，黄雀在后”，Lotp-L15 可以锁定别人，这些位本身又被三个 BL 位锁定，BL15-10 用于锁定 L15-L10，BL9-4 用于锁定 L9-L4，BLotp 用于锁定 Lotp。所有的这 16 个锁定位也具有 OTP 特性，通俗的讲就是这些“锁”没有“钥匙”，一旦锁死就再也改不回来了，所以锁定时一定要小心。

Mifare UltraLight 的读写操作和 Mifare S50 是完全兼容的，这里的“兼容”是指二者可以使用同一个读卡器硬件，同一套软件。当然若软硬件完全相同就不是两种卡了，二者的区别主要体现在软件操作上，包括以下 4 个方面：

一是 Mifare UltraLight 的卡序列号有 7 个字节，而 Mifare S50 的卡序列号只有 4 个字节，因此在卡片防冲突选择阶段需要两层(Cascade，93H 和 95H)操作；

二是 Mifare UltraLight 没有密码，不需要验证；

三是 Mifare UltraLight 的 Page 相当于 Mifare S50 的 BLOCK，因此 Mifare UltraLight 有

16 个 BLOCK，且每个 BLOCK 只有 4 个字节，而 Mifare S50 有 64 个 BLOCK,每个 BLOCK 有 16 个字节，也就是说 Mifare UltraLight 卡每次只能写 4 个字节，而 Mifare S50 每次可以写 16 个字节信息，读都是可以一次读 16 个字节信息；

四是 Mifare UltraLight 没有电子钱包功能。

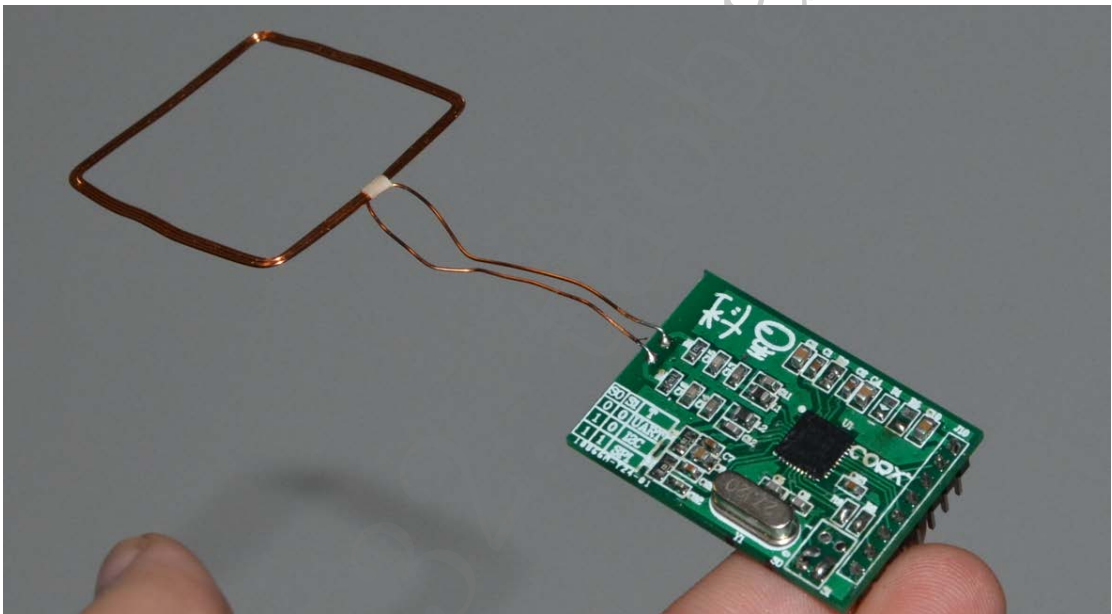
基于以上四点，在 Mifare S50 的程序中在卡请求命令成功执行后，如果判断卡类型字节为 Mifare UltraLight(0044H),则在之后的操作中增加第二层防冲突选择，卡选择成功后直接对卡片的 0-15 块进行读写操作，每次读写只关注前 4 个字节，不使用电子钱包功能，这样就可以两种卡片完全兼容了。

二、NFC 模块介绍

1、科星 NFC 模块可以有三种接口与外界通信，分别是 UART、SPI 和 I2C。三种接口的选择是靠 S0 和 S1 信号控制的，S0 或 S1 接高电平或低电平可以选择是哪种接口与外界通信，如下表所示：

	S0	S1
UART	0	0
I2C	1	0
SPI	1	1

如下图所示，NFC 模块引出了 8 个引脚与外界连接：



引脚定义如下表（标 J10 的一端为 1 脚）：

序号	名称	引脚作用
1	3.3V	3.3V 电源
2	SCK	SPI 时钟信号
3	MISO	SPI 接口时主器件数据输入，从器件数据输出
4	MOSI/SDA/Tx	SPI 接口时主器件数据输出，从器件数据输入 / IIC 接口时 SDA / UART 时的 Tx
5	NSS/SCL/Rx	SPI 接口时从器件使能信号 / IIC 接口时 SCL / UART 时的 Rx
6	RSTOUT_N	复位信号
7	IRQ	中断信号，NFC 只能是从设备，不能主动发数据给 MCU，这个终端信号可以告诉 MCU，有数据来了
8	GND	地

6.1.2 P70_IRQ pin

In addition to the physical link used to communicate with the host controller, another dedicated IRQ line is used (see reference [3]) to inform the host controller when a response to a command is available.

This IRQ pin is driven automatically by the firmware. It is used by the handshake mechanism.

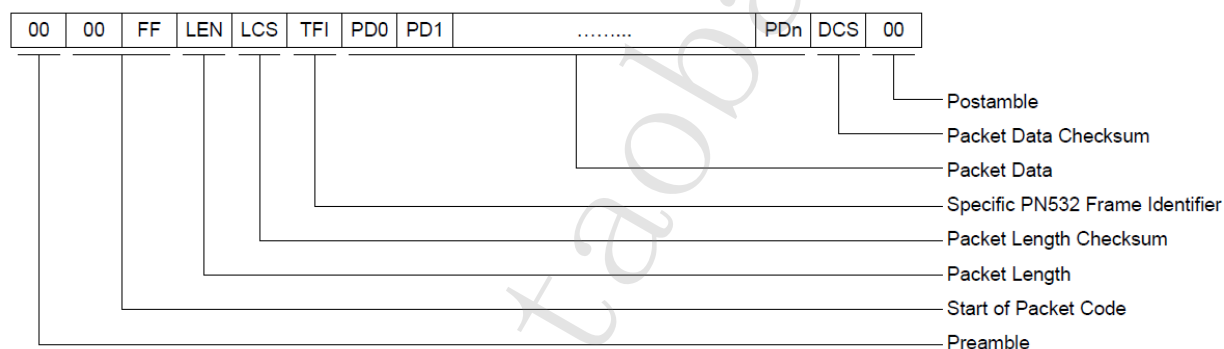
三、NFC 模块读写 Mifare 卡

3.1、上位机读写卡测试

3.1.1、硬件连接

NFC 模块选择 UART 模式，但是这个模块输出的是 TTL 电平的，所以需要用一个 TTL 转 USB 的小模块信号转换一下就可以直接插在电脑的 USB 口了。

3.1.2、指令帧格式



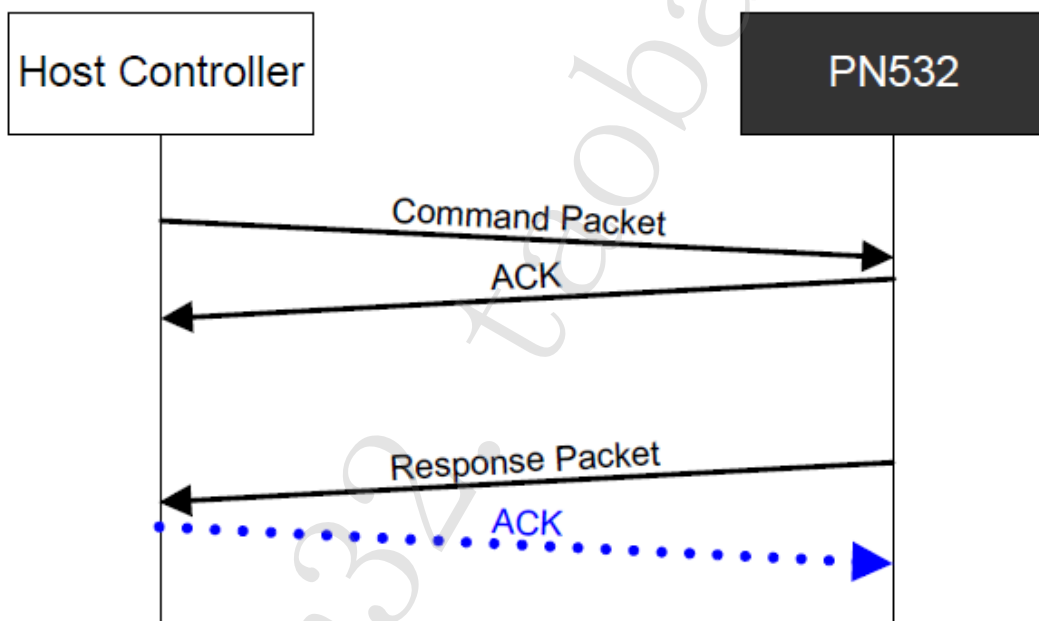
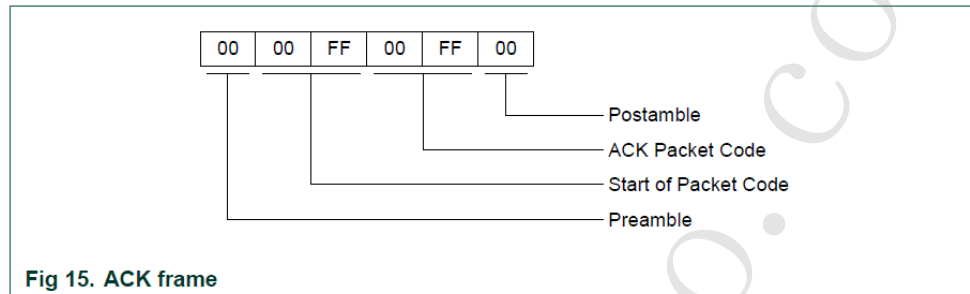
- **PREAMBLE** 1 byte⁴,
- **START CODE** 2 bytes (0x00 and 0xFF),
- **LEN** 1 byte indicating the number of bytes in the data field (TFI and PD0 to PDn),
- **LCS** 1 Packet Length Checksum LCS byte that satisfies the relation:
Lower byte of [LEN + LCS] = 0x00,
- **TFI** 1 byte frame identifier, the value of this byte depends on the way of the message
 - **D4h** in case of a frame from the host controller to the PN532,
 - **D5h** in case of a frame from the PN532 to the host controller.
- **DATA** LEN-1 bytes of Packet Data Information
The first byte PD0 is the Command Code,
- **DCS** 1 Data Checksum DCS byte that satisfies the relation:
Lower byte of [TFI + PD0 + PD1 + ... + PDn + DCS] = 0x00,
- **POSTAMBLE** 1 byte².

6.2.1.3 ACK frame

The specific ACK frame is used for the synchronization of the packets and also for the abort mechanism.

This frame may be used either from the host controller to the PN532 or from the PN532 to the host controller to indicate that the previous frame has been successfully received.

ACK frame:



3.1.3、唤醒 NFC 模块

NFC 模块开始时处于休眠状态，需要将其初始化到 normal 状态，此指令不需要放卡，而且每次上电后唤醒一次就可以。但是如果发送休眠指令后，还是需要唤醒。

指令如下：

PC→PN532:

55 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 03 FD D4 14 01 17 00

PN532→PC:

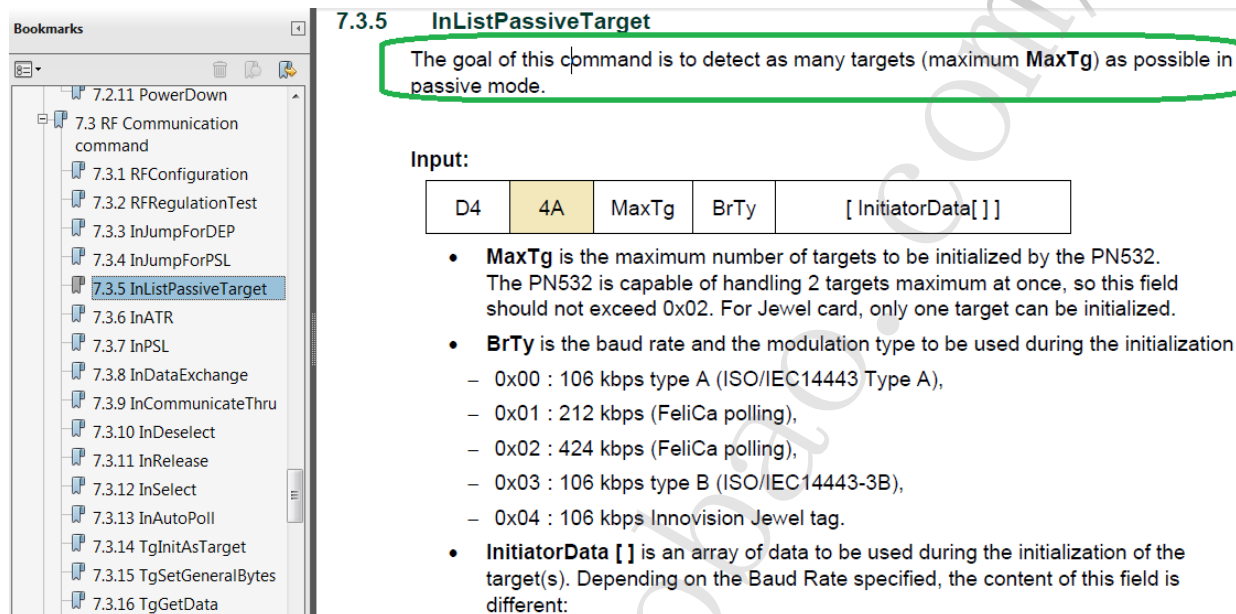
00 00 FF 00 FF 00

PN532→PC:

00 00 FF 02 FE D5 15 16 00

3.1.4、寻找 Mifare 卡（也可以叫 NFC 标签）

PN532 手册对这个指令有详细的介绍，如下图所示：



7.3.5 InListPassiveTarget

The goal of this command is to detect as many targets (maximum **MaxTg**) as possible in passive mode.

Input:

D4	4A	MaxTg	BrTy	[InitiatorData[]]
----	----	-------	------	----------------------

- MaxTg** is the maximum number of targets to be initialized by the PN532. The PN532 is capable of handling 2 targets maximum at once, so this field should not exceed 0x02. For Jewel card, only one target can be initialized.
- BrTy** is the baud rate and the modulation type to be used during the initialization
 - 0x00 : 106 kbps type A (ISO/IEC14443 Type A),
 - 0x01 : 212 kbps (FeliCa polling),
 - 0x02 : 424 kbps (FeliCa polling),
 - 0x03 : 106 kbps type B (ISO/IEC14443-3B),
 - 0x04 : 106 kbps Innovision Jewel tag.
- InitiatorData []** is an array of data to be used during the initialization of the target(s). Depending on the Baud Rate specified, the content of this field is different:

指令如下：

PC→PN532:

00 00 FF 04 FC D4 4A 01 00 E1 00

PN532→PC:

00 00 FF 00 FF 00

PN532→PC:

00 00 FF 0C F4 D5 4B 01 01 00 04 08 04 94 8A 3B 0B 6A 00

下面解释一下指令

00 00 FF 04 FC D4 4A 01 00 E1 00

00 报头

00 FF 开始包指令

04 包长度

FC 包长度校验(04+FC=0x100)

D4 4A 指令的标识码

01 寻卡的数量, 0~2

00 0x00 : 106 kbps type A (ISO/IEC14443 Type A),

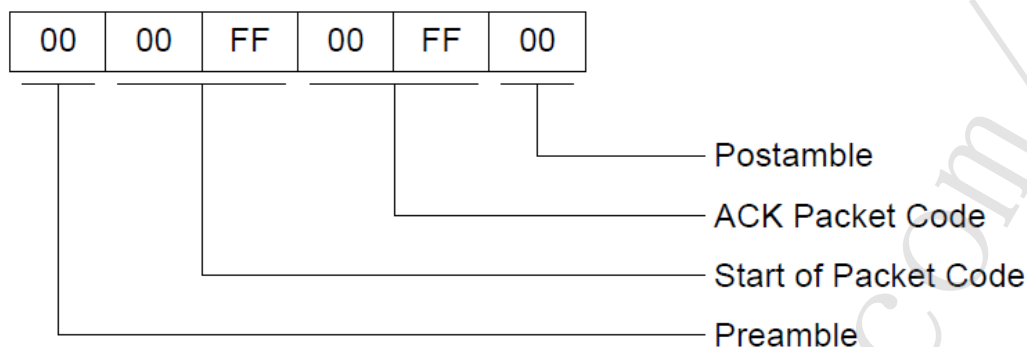
E1 数据校验和 (0x100- (0xD4+0x4A+0x01+0x00))

00 报尾

PN532→PC:

00 00 FF 00 FF 00

是 ACK 应答帧



PN532→PC:

00 00 FF 0C F4 D5 4B 01 01 00 04 08 04 94 8A 3B 0B 6A 00

00 报头

00 FF 开始包指令

0C 包长度

F4 包长度校验

D5 4B 指令的标识码(是主机发送的指令 加 1, 主机发的 D4 4A, 回 D5 4B)

01 目标卡 1

01 找到目标卡的数量

00 04 ATQA 说明是 Mifare S50 卡, §1.3 节详细讲述了, Mifare 卡的分类

08 卡的容量 08=1K

04 UID 的字节数

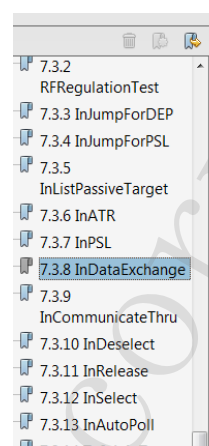
94 8A 3B 0B 卡的 UID

6A (0x100—累加和) 校验

00 报尾

3.1.5、密码验证（授权）

使用的指令如下：



7.3.8 InDataExchange

This command is used to support protocol data exchanges between the PN532 as initiator and a target.

Input:

D4	40	Tg	[DataOut []]
----	----	----	----------------

- **Tg** is a byte containing the logical number of the relevant target. This byte contains also a *More Information* (MI) bit (bit 6) indicating, when set to 1, that the host controller wants to send more data that all the data contained in the **DataOut []** array (see Chaining mechanism §7.4.5, p: 178). This bit is only valid for a TPE target.
- **DataOut** is an array of raw data (from 0 up to 262 bytes) to be sent to the target by the PN532 (see §7.4.7, p: 186).

The Mifare specific command byte **Cmd** may take one of the possible values:

0x60 / 0x61	Authentication A / Authentication B
0x30	16 bytes reading
0xA0	16 bytes writing
0xA2	4 bytes writing
0xC1	Incrementation
0xC0	Decrementation
0xB0	Transfer
0xC2	Restore

这里我们使用 0x60, (KEY A 验证) 进行密码验证, KEY A 和 KEY B 的默认密码一般都是 6 个 FF, FF FF FF FF FF FF。

注意: 这里的验证只针对 Mifare S50 卡, Mifare UltraLight 卡无需密码验证

下面我们先看指令:

PC→PN532:

00 00 FF F F1 D4 40 01 60 03 FF FF FF FF FF FF 94 8A 3B 0B 2A 0

PN532→PC:

00 00 FF 00 FF 00

PN532→PC:

00 00 FF 03 FD D5 41 00 EA 00

D4 40 指令的标识码

01 数据大于 6 个字节就设置为 1, 就是告诉卡, 主机要发送比较多的数据了

This byte contains also a *More Information* (MI) bit (bit 6) indicating, when set to 1, that the host controller wants to send more data that all the data contained in the **DataOut []** array (see Chaining mechanism §7.4.5, p: 178). This bit is only valid for a TPE target.

60 验证密码 A

03 在第三块存贮密码

FF FF FF FF FF FF 密码 A 默认为 6 个 FF

94 8A 3B 0B 要验证的 Mifare 卡的 UID

2A 校验

PN532→PC:

00 00 FF 03 FD D5 41 00 EA 00

00

00 FF

03 FD

D5 41 指令的标识码 (发送的标识码 加 1)

00 返回的操作标志, 00 表示成功, 错误号所代表的意思在 PN532 的手册可以查到, 如下图:

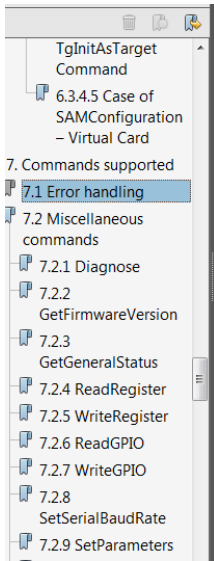


Table 13. Error code list

Error cause	Error code
Time Out, the target has not answered	0x01
A CRC error has been detected by the CIU	0x02
A Parity error has been detected by the CIU	0x03
During an anti-collision/select operation (ISO/IEC14443-3 Type A and ISO/IEC18092 106 kbps passive mode), an erroneous Bit Count has been detected	0x04
Framing error during Mifare operation	0x05
An abnormal bit-collision has been detected during bit wise anti-collision at 106 kbps	0x06
Communication buffer size insufficient	0x07
RF Buffer overflow has been detected by the CIU (bit BufferOvfl of the register <i>CIU_Error</i>)	0x09
In active communication mode, the RF field has not been switched on in time by the counterpart (as defined in NFCIP-1 standard)	0x0A

EA 校验
00 报尾

3.1.6、Mifare 卡数据存储格式

1) Mifare S50 卡

前面讲到了，S50 卡有 1K 的空间，分为 16 个扇区，每个扇区分为 4 块，每块 16 字节。

A.MF1 S50 卡分为 16 个扇区，每个扇区由 4 块（块 0、块 1、块 2、块 3）组成，（我们也将 16 个扇区的 64 个块按绝对地址编号为 0~63，存贮结构如下图所示：

扇区 0	块 0		数据块	0
	块 1		数据块	1
	块 2		数据块	2
	块 3	密码 A 存取控制 密码 B	控制块	3
扇区 1	块 0		数据块	4
	块 1		数据块	5
	块 2		数据块	6
	块 3	密码 A 存取控制 密码 B	控制块	7
		:		
		:		
		:		
扇区 15	0		数据块	60
	1		数据块	61
	2		数据块	62
	3	密码 A 存取控制 密码 B	控制块	63

B.第 0 扇区的块 0（即绝对地址 0 块），它用于存放厂商代码，已经固化，不可更改。

C.每个扇区的块 0、块 1、块 2 为数据块，可用于存贮数据。

数据块可作两种应用：

用作一般的数据保存，可以进行读、写操作。

用作数据值，可以进行初始化值、加值、减值、读值操作。

D.每个扇区的块 3 为控制块，包括了密码 A、存取控制、密码 B。具体结构如下：

A0 A1 A2 A3 A4 A5	FF 07 80 69	B0 B1 B2 B3 B4 B5
-------------------	-------------	-------------------

密码 A（6 字节） 存取控制（4 字节） 密码 B（6 字节）

这里的密码 A 和密码 B 默认都是 6 个 FF，这里密码 B 可以读取出来，密码 A 永远读取不出来。

E.每个扇区的密码和存取控制都是独立的，可以根据实际需要设定各自的密码及存取控制。存取控制为 4 个字节，共 32 位，扇区中的每个块（包括数据块和控制块）的存取条件是由密码和存取控制共同决定的，在存取控制中每个块都有相应的三个控制位，定义如下：

块 0: C10 C20 C30
 块 1: C11 C21 C31
 块 2: C12 C22 C32
 块 3: C13 C23 C33

三个控制位以正和反两种形式存在于存取控制字节中，决定了该块的访问权限（如进行减值得操作必须验证 KEY A，进行加值得操作必须验证 KEY B，等等）。三个控制位在存取控制字节中的位置，以块 0 为例：

表 3-1.对块 0 的控制：

Bit	7	6	5	4	3	2	1	0
字节 6				C20_b				C10_b
字节 7				C10				C30_b
字节 8				C30				C20
字节 9								

（注：C10_b 表示 C10 取反）

表 3-2. 存取控制表（4 字节，其中字节 9 为备用字节）：

bit	7	6	5	4	3	2	1	0
字节 6	C23_b	C22_b	C21_b	C20_b	C13_b	C12_b	C11_b	C10_b
字节 7	C13	C12	C11	C10	C33_b	C32_b	C31_b	C30_b
字节 8	C33	C32	C31	C30	C23	C22	C21	C20
字节 9								

（注：_b 表示取反）

F.数据块（块 0、块 1、块 2）的存取控制如下：

表 3-3 .数据块（块 0、块 1、块 2）的存取控制权限表

控制位 (X=0..2)			访 问 条 件 （对数据块 0、1、2）			
C1X	C2X	C3X	Read	Write	Increment	Decrement, transfer, Restore
0	0	0	KeyA B	KeyA B	KeyA B	KeyA B
0	1	0	KeyA B	Never	Never	Never
1	0	0	KeyA B	KeyB	Never	Never
1	1	0	KeyA B	KeyB	KeyB	KeyA B
0	0	1	KeyA B	Never	Never	KeyA B
0	1	1	KeyB	KeyB	Never	Never
1	0	1	KeyB	Never	Never	Never
1	1	1	Never	Never	Never	Never

（KeyA|B 表示密码 A 或密码 B，Never 表示任何条件下不能实现）

例如：当块 0 的存取控制位 C10 C20 C30=1 0 0 时，验证密码 A 或密码 B 正确后可读；验证密码 B 正确后可写；不能进行加值、减值得操作。

G.控制块块 3 的存取控制与数据块（块 0、1、2）不同，它的存取控制如下：

表 3-4 .数据块（块 3）的存取控制权限表

			密码 A		存取控制		密码 B	
C13	C23	C33	Read	Write	Read	Write	Read	Write
0	0	0	Never	KeyA B	KeyA B	Never	KeyA B	KeyA B
0	1	0	Never	Never	KeyA B	Never	KeyA B	Never
1	0	0	Never	KeyB	KeyA B	Never	Never	KeyB
1	1	0	Never	Never	KeyA B	Never	Never	Never
0	0	1	Never	KeyA B	KeyA B	KeyA B	KeyA B	KeyA B
0	1	1	Never	KeyB	KeyA B	KeyB	Never	KeyB
1	0	1	Never	Never	KeyA B	KeyB	Never	Never
1	1	1	Never	Never	KeyA B	Never	Never	Never

例如：当块 3 的存取控制位 C13 C23 C33=1 0 0 时，表示：

密码 A：不可读，验证 KEYA 或 KEYB 正确后，可写（更改）。

存取控制：验证 KEYA 或 KEYB 正确后，可读、可写。

密码 B：验证 KEYA 或 KEYB 正确后，可读、可写。

举例说明：S50 卡修改各区块控制位值和数据

以出厂设置“FF 07 80 69”控制条件为例，计算分析它的访问权限。

1)对“FF 07 80 69”值进行计算，该值定位于各区块 3 的 6、7、8、9 四个字节内，字节 6=FF，字节 7=07，字节 8=80，字节 9=69（备用字节，不予计算）。

2)例如：字节 6=FF，对应其二进制值=1111 1111，则对 6、7、8 这三个字节进行二进制转换结果见下表：

字节 6 = 1111 1111	字节 7 = 0000 0111	字节 8 = 1000 0000
------------------	------------------	------------------

3) 参照表 3-2 中的算法, 字节 6 的全部二进制值取反, 字节 7 的低四位二进制值取反, 字节 8 不变, 得到:

字节号	对应二进制值	位置	高 4 位	位置	低 4 位
字节 6	1111 1111	C2Y	0 0 0 0	C1Y	0 0 0 0
字节 7	0000 0111	C1Y	0 0 0 0	C3Y	1 0 0 0
字节 8	1000 0000	C3Y	1 0 0 0	C2Y	0 0 0 0
所属块位			块 3 块 2 块 1 块 0		块 3 块 2 块 1 块 0

4) 对以上 6、7、8 字节的存取/控制二进制已取反值, 依照上表块位转换为各块控制值, 如下表:

块 3 位	字节 7、字节 6、字节 8 = C13、C23、C33 = C1Y、C2Y、C3Y = 0 0 1
块 2 位	字节 7、字节 6、字节 8 = C12、C22、C32 = C1Y、C2Y、C3Y = 0 0 0
块 1 位	字节 7、字节 6、字节 8 = C11、C21、C31 = C1Y、C2Y、C3Y = 0 0 0
块 0 位	字节 7、字节 6、字节 8 = C10、C20、C30 = C1Y、C2Y、C3Y = 0 0 0

注意: 高 4 位的各块值=低 4 位的各块值时, 其值可用。高 4 位值≠低 4 位值时, 其值不可用!

5) 查对访问权限 (块 0、1、2 存取控制依照表 3-3, 块 3 存取控制依照表 3-4), 该例 “FF 07 80 69” 的访问权限为:

- ◆ 块 3 = 001: 权限为: KeyA 不可读, 验证 KeyA 或验证 KeyB 正确后 KeyB 可读; 验证 KeyA 或验证 KeyB 正确后可改写 KeyA 和 KeyB, 验证 KeyA 或 KeyB 正确后可读也可写 “控制位”。
- ◆ 块 2 = 块 1 = 块 0 = 000: 权限为: 验证 KeyA 或 KeyB 后可读或写该块数据, 也可进行加值、减值以及初始化值的操作。

3.1.7、写数据指令

能否写的话，S50 卡看 权限控制了，或是需要验证 KEYA，或是验证 KEYB，或是只读的，看 3.1.6 节了，对权限这里写的很详细。

写指令有两种，如下图：

0xA0	16 bytes writing
0xA2	4 bytes writing

S50 卡支持一次写 16 字节数据和一次写 4 字节数据；但是 UltraLight 卡只支持一次写 4 字节数据。不过我测试过一次往 UltraLight 卡写入 16 字节数据，可以写成功，但是只有前 4 个字节写入了，其余的 12 字节数据不变。

写数据指令：

PC→PN532:

00 00 FF 15 EB D4 40 01 A0 02 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
D1 00

PN532→PC:

00 00 FF 00 FF 00

PN532→PC:

00 00 FF 03 FD D5 41 00 EA 00

D4 40 指令标识符

01 大于 6 字节

A0 16 字节写数据指令

02 往第 02 块（三块）写

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 要写入的 16 字节数据

D5 41 返回的指令标示符

00 说明写入成功

PC→PN532:

00 00 FF 09 F7 D4 40 01 A2 04 00 01 02 03 3F 00 写入 4 字节的指令

3.1.8、读数据指令

能否读的话，S50 卡看 权限控制了，或是需要验证 KEYA，或是验证 KEYB，看 3.1.6 节了，对权限这里写的很详细。

指令如下：

PC→PN532:

00 00 FF 05 FB D4 40 01 30 02 B9 00

PN532→PC:

00 00 FF 00 FF 00

00 00 FF 13 ED D5 41 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 72 00

D4 40

01

30 读 16 字节

01 读 02 块的数据，也就是第三块，从 0 开始数的

D5 41

00 读取成功

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 读取的数据